动态选择策略的和声教与学混合算法 *

黎延海，拓守恒，雍龙泉

(陕西理工大学 数学与计算机科学学院，陕西 汉中 723001)

摘　要： 为提高对多种不同类型问题的优化性能，提出了一种基于和声搜索和教与学优化的混合优化算法（HHSTL）。在不同的进化阶段，HHSTL 算法依据种群活跃率及种群最优个体更新率动态地确定和声算法或教与学算法作为下一周期种群更新方式的比例，并在标准教与学算法中增加了“自学”策略来提高算法的全局寻优能力。通过 16 个不同类型的 Benchmark 函数进行仿真，并与 7 种优秀算法进行结果比较及 Wilcoxon 秩和检验分析，结果表明 HHSTL 算法汲取了和声搜索和教与学优化算法的优点，具有求解精度高、稳定性好等特点，能够求解更多的较为复杂的优化问题。

关键词： 和声搜索；教与学优化；动态选择策略；“自学”策略

中图分类号： TP301.6　　doi: 10.3969/j.issn.1001-3695.2018.06.0399

Hybrid algorithm based on harmony search and teaching-learning-based optimization using dynamic selection strategies

Li Yanhai, Tuo Shouheng, Yong Longquan

(*School of Mathematics & Computer Science Shaanxi University of Technology*, *Hanzhong Shaanxi* 723001, *China*)

Abstract: This paper presents a hybrid algorithm based on Harmony Search and Teaching-Learning-based Optimization (HHSTL) for solving various types of optimization problems. In different evolution stages, HHSTL algorithm dynamically determines the proportion using the harmony search (HS) or Teaching-Learning-based optimization (TLBO) algorithm as the means of updating the next generation of population on basis of the population activity rate and population optimal individual renewal rate, in addition , it proposed a "self-study" strategy to improve the global optimization performance of the TLBO algorithm. It employed 16 various types of Benchmark functions to investigate the performance of HHSTL, and compared with seven algorithms by Wilcoxon rank sum tests . The experimental results indicate that HHSTL algorithm adopted merits of HS and TLBO, and can solve more complex optimization problems because of its high solution precision and excellent stability.

Key words: harmony search; teaching-learning-based optimization; dynamic selection strategies; “self-study” strategy

在科学计算和工程技术中，经常会遇到类型各异的优化问题，单一的优化算法往往具有偏好性，适用的范围有限，因此，将具有不同优点的算法进行融合能够扩大求解问题的范围，具有重要的实际意义。和声搜索算法（harmony search，HS）[1,2]是 Geem 等人通过模拟现实中的乐队即兴创作过程而于 2001 年提出的一种启发式优化算法。HS 算法具有较强的全局搜索能力，适用于多模态优化问题的求解，但局部开发能力较弱，使得在优化单模态问题时所得到的解的精度不高[3]。教与学优化算法（teaching-learning-based optimization，TLBO）是 Rao 等人[4,5]受课堂教学过程启发在 2011 年提出的一种新的群体智能优化算法。TLBO 算法具有很强的收敛能力，已经被证明在很多标准优化函数及工程问题上相较于其他许多全局优化算法具有更好的表现，是一种有效的全局优化方法。但研究揭示[6]该算法在“教”阶段存在固有的起源偏差，对以坐标原点为最优解的问题有着原始的偏好，但对最优解不在原点的问题则优化效果不太理想。这两种算法具有不同的特点，适合解决的问题类型也各不相同，因此将两种算法进行融合，能达到优势互补，各取所长的效果。目前，关于将这两种算法相融合的研究主要有 Aydogdu 等人[7]将 HS 策略融入 TLBO 算法而提出了混合算法，并应用于钢架结构设计问题；Tuo 等人[8]提出了面向高维复杂优化问题的 HSTLBO 算法，算法应用了一个自适应策略对两种算法进行了混合；Yong 等人[9]将 TLBO 算法作为即兴创作新和声的一种策略，将两种算法进行了混合，并应用于非负线性最小二乘问题的求解；Ouyang 等人[10]使用 HS 的即兴创作过程来优化“教阶段”的最优个体，而提出了 HTLBO-HS 算法。现有这些混合算法在一定程度上提高了算法的性能，但并没有从

收稿日期： 2018-06-20；**修回日期：** 2018-08-14　**基金项目：** 国家自然科学基金资助项目（11502132）；陕西省教育厅科研资助项目（14JK1130）；陕西理工大学校级科研项目（SLGKY2017-05）

作者简介： 黎延海（1981-），男，陕西安康人，讲师，硕士，主要研究方向为智能优化算法（liyanhai@snut.edu.cn）；拓守恒（1978-），男，副教授，博士，主要研究方向为智能优化算法、生物信息分析与处理；雍龙泉（1980-），男，教授，博士，主要研究方向为优化理论与算法设计、智能优化算法.

根本上克服它们的固有缺点。使用合适的融合方式将两种算法进行混合，才能充分发挥它们各自的优点，取得优势互补的效果。为此，本文提出了一种和声教与学混合算法（hybrid algorithm based on HS and TLBO , HHSTL）。HHSTL 算法依据种群活跃率及种群最优个体更新率这两种机制动态地确定 HS 算法或 TLBO 算法在下一周期中作为种群更新方式的比例。利用国际上通用的 16 个不同类型的 Benchmark 测试函数[11,12] 进行仿真检验，并与七种优化算法（DIHS[13]、IHS[14]、RCGHS[15]、TLBO、ITLBO[16]、MATLBO[17]、TLBO-GC[18]）进行了比较，检验了所提算法的性能。

1　和声搜索算法

1.1　标准和声搜索算法

标准和声算法的基本步骤[1]如下：

a）参数设置。

设置问题的维数 D，和声记忆库的大小 HMS，最大迭代次数 $T_{\max}$，记忆库取舍概率 $HMCR$，局部微调概率 PAR，局部微调步长 Bw。

b）随机初始化和声记忆库 HM。

$$HM=\begin{bmatrix} x^1 \\ x^2 \\ \cdots \\ x^{HMS} \end{bmatrix}=\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_D^1 \\ x_1^2 & x_2^2 & \cdots & x_D^2 \\ \cdots & \cdots & \cdots & \cdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_D^{HMS} \end{bmatrix}$$

c）创作一个新的和声。

每次迭代可产生一个新和声 X^{new}。新和声的每一个音调 $x_i^{new}\ (i=1,2,\cdots,D)$ 按以下三种规则产生：①以概率 $HMCR$ 从和声记忆库 HM 的第 i 维 $[x_i^1,x_i^2,\cdots,x_i^{HMS}]$ 中选择；②将①中产生的 x_i^{new} 以概率 PAR 进行微调，方法如下：

$$x_i^{new}=x_i^{new}\pm rand\cdot Bw(i) \tag{1}$$

③以概率 $1-HMCR$ 在搜索空间内随机生成，具体如下：

$$x_i^{new}=xL_i+rand\cdot(xU_i-xL_i) \tag{2}$$

d）更新和声记忆库。

对步骤 c）中产生的新和声 X^{new} 进行评估，如果优于 HM 中的最差和声 X^{worst}，则用 X^{new} 替换 X^{worst}，否则转 c）。重复步骤 c）d），直到满足终止条件停止。

1.2　改进的和声算法

为改进和声算法的性能，文献[14]提出了一种改进的和声算法（IHS），算法对参数 PAR 和 Bw 进行动态地调整。参数 PAR 随迭代的增加而线性增大，Bw 呈指数地递减。

$$PAR(t)=PAR_{\min}+\frac{PAR_{\max}-PAR_{\min}}{T_{\max}}\times t \tag{3}$$

$$Bw(t)=Bw_{\max}\exp\left(\frac{\ln\left(\frac{Bw_{\min}}{Bw_{\max}}\right)}{T_{\max}}\times t\right) \tag{4}$$

其中：$T_{\max}$ 表示最大迭代次数；t 为当前迭代次数；$PAR_{\max}$、$PAR_{\min}$ 分别为最大、最小微调概率；$Bw_{\max}$、$Bw_{\min}$ 分别为最大、最小微调步长。

2　教与学优化算法

2.1　标准的教与学优化算法

标准 TLBO 算法模拟了班级的教学和学习行为，主要包括“教”阶段和“学”阶段两个阶段。“教”阶段，老师向每位学员传授知识，提高班级的整体水平；“学”阶段，学员之间相互学习，取长补短，共同进步。基本步骤[4]如下：

a) 参数设置。

NP 表示班级的人数(种群的规模), D 表示学习的科目数(问题的维数), $T_{\max}$ 为最大迭代次数。

b) 在解空间随机初始化 NP 位学员：

$$X^j=XL+rand\cdot(XL-XU) \tag{5}$$

其中：$j=1,2,\cdots,NP$；$rand$ 为 (0,1) 间的随机数。

c) “教”阶段。

$$X_{new}^j=X_{old}^j+difference \tag{6}$$

$$difference=rand(1,D)\times\left(X^{teacher}-T_F\cdot X^{mean}\right) \tag{7}$$

其中：X_{old}^j、X_{new}^j 分别表示第 j 位学员 X^j 在教学前、后的知识水平；$X^{teacher}$ 为老师（即适应度值最优的学员）；$T_F=round(1+rand(0,1))$ 表示教学因子，取值为 1 或 2；$X^{mean}=\frac{1}{NP}\sum_{j=1}^{NP}X^j$ 为班级学员知识水平的均值。

适应度值 $f(X^j)$ 反映了学员 X^j 的学习成绩，教学完成后，如果 $f(X_{new}^j)<f(X_{old}^j)$，则更新个体，即 $X^j=X_{new}^j$，否则不更新。

d) “学”阶段。

学员 X^j 从班级中随机抽取一个学员 X^r，$r=1,2,\cdots,NP$，$r\neq j$，进行交流学习。

$$X_{new}^j=\begin{cases} X_{old}^j+rand\cdot(X^r-X^j) & f(X^r)<f(X^j) \\ X_{old}^j+rand\cdot(X^j-X^r) & f(X^j)<f(X^r) \end{cases} \tag{8}$$

学习完成后，如果 $f(X_{new}^j)<f(X_{old}^j)$，则更新个体，即 $X^j=X_{new}^j$，否则不更新。

e) 如果满足终止条件，则优化结束并输出最优解，否则转至 c）继续。

2.2　教与学优化算法的改进

TLBO 算法具有较强的收敛能力，但易于陷入局部最优，尤其对中心旋转偏移的复杂问题，存在固有的缺陷。为增强算法的寻优性能，本文做了如下改进：

a）班级中的最差成绩往往容易拉低班级的平均成绩，故用班级中的成绩最差者 X^{worst} 去替换班级平均水平 X^{mean}，可以尽快提升班级的整体水平，即

$$difference=rand(1,D)\times\left(X^{teacher}-T_F\cdot X^{worst}\right) \tag{9}$$

b）学生学习时，不仅会在同学之间进行交流，也会向老师求教，故对学习方式做如下改进：

$$X_{new}^{j}=\begin{cases}X_{old}^{j}+F\cdot[\mu(X^{r}-X^{j})+(1-\mu)(X^{teacher}-X^{j})] & f(X^{r})<f(X^{j})\\ X_{old}^{j}+F\cdot[\mu(X^{j}-X^{r})+(1-\mu)(X^{teacher}-X^{j})] & f(X^{j})<f(X^{r})\end{cases} \tag{10}$$

$$\mu=1-\left(\frac{t}{T_{\max}}\right)^{\frac{1}{2}} \tag{11}$$

μ 的值随迭代的增加而逐渐减小，表明在迭代的前期，同学间的交流较多，但随着迭代的进行，学生会更多地向老师请教。F 表示学习因子，取值于 $(0,1)$ 间。

c）学生在学习之余，也会根据自己当前的学习状况，挑选部分感兴趣的课程进行“自学”，从而提升自己的学业水平。具体如下：对第 i 门课程，产生一个 $(0,1)$ 间的随机数 $rand$，用于不同的“自学”方式。

$$x_{new,i}^{j}=\begin{cases}x_{old,i}^{j}+2\times(rand-0.5)\cdot\lambda & if \quad rand<SCR\\ x_i^{L}+\text{rand}\times\left(x_i^{U}-x_i^{L}\right) & elseif \quad rand<SDR\\ x_{old,i}^{j} & otherwise\end{cases} \tag{12}$$

$$\lambda=\lambda_{\max}\times\left(\frac{\lambda_{\min}}{\lambda_{\max}}\right)^{\left(\frac{t}{T_{\max}}\right)^{2}} \tag{13}$$

$i=1,\cdots D, j=1,\cdots,NP$。其中：SCR（self-culture rate, SCR）为自学概率；SDR（self-development rate, SDR）为自我拓展率；λ 表示自学步长，$\lambda_{\max}$、$\lambda_{\min}$ 分别为最大、最小步长。

对每一门课程，学生以 SCR 的概率进行自我调整，以 SDR 的概率在解空间随机搜素，进行自我拓展学习。为平衡算法在搜索过程中的空间扰动能力和局部寻优能力，设计自学步长随迭代的进行由 $\lambda_{\max}$ 逐步动态缩小为 $\lambda_{\min}$。

3　“和声”-“教与学”混合优化算法

对不同的优化目标，HS 和 TLBO 算法所表现的优化性能各不相同，为充分发挥它们各自的优点，更有效地解决不同类型的优化问题，本文设计了一种动态选择策略，利用种群活跃率（AR）及种群最优个体更新率(BR)两种机制对两种算法进行了融合，动态地确定在下一周期中选择 HS 算法或 TLBO 算法作为种群更新方式的比例。

3.1　动态选择策略

算法每经过 T（选择周期）次迭代后使用 DP（动态选择概率）来重新确定两种算法的选择比例。在第 k 个选择周期，每次迭代首先生成一个随机数 $r^{(k)}\in(0,1)$。如果 $r^{(k)}<DP^{(k)}$，选择 HS 算法来更新种群；否则，选择 TLBO 算法来更新种群。

种群活跃率（AR）= 更新成功个体数/产生新个体总数

种群最优个体更新率（BR）= 种群最优个体更新次数/所产生种群的数目

当 $k=1$ 时，令 $DP^{(1)}=0.5$，对 $k=2,\cdots,\left[\frac{T_{\max}}{T}\right]$

$$AR_H^{(k)}=\frac{c_1(k)}{NP\times T\times DP^{(k-1)}},\ AR_T^{(k)}=\frac{c_2(k)}{NP\times T\times(1-DP^{(k-1)})},$$

$$BR_H^{(k)}=\frac{b_1(k)}{T\times DP^{(k-1)}},\ BR_T^{(k)}=\frac{b_2(k)}{T\times(1-DP^{(k-1)})},$$

$$DP^{(k)}=\frac{AR_H^{(k)}}{AR_H^{(k)}+AR_T^{(k)}}+\frac{BR_H^{(k)}}{BR_H^{(k)}+BR_T^{(k)}} \tag{14}$$

其中：$T_{\max}$ 为最大迭代次数；$c_1(k)$、$c_2(k)$ 表示第 k 个选择周期内分别使用 HS 和 TLBO 算法更新成功的个体数目；$b_1(k)$、$b_2(k)$ 表示第 k 个选择周期内分别使用 HS 和 TLBO 算法更新种群最优个体的次数；$AR_H^{(k)}$、$AR_T^{(k)}$ 分别表示第 k 个周期内使用 HS 和 TLBO 算法所产生种群的活跃率；$BR_H^{(k)}$、$BR_T^{(k)}$ 分别表示第 k 个周期内使用 HS 和 TLBO 算法所产生种群的最优个体更新率；$DP^{(k)}$ 为第 k 个周期的动态选择概率。

在第一个周期，两种算法以相同的概率被选择，公平竞争；在后面的每个周期，使用动态选择概率来决定两种算法的选择比例，哪种算法在迭代过程中更活跃，并且对算法寻优的贡献越大，被选择的概率也就越大。迭代过程中，种群更新成功个体的数目越多，表明算法越活跃，适用于所求解的问题，但算法活跃并不一定能使得种群的最优解得到更新，所以使用由种群活跃率和最优个体更新率共同确定的动态选择概率来决定新一代种群的更新方式，动态地确定两种算法的混合比例以适应不同的优化对象，能够充分发挥两种算法的长处，提高算法的寻优能力。针对不同的优化问题和同一问题的不同阶段，选择策略能够根据两种算法的具体表现动态地进行调节，充分发挥 HS 算法优秀的全局搜索能力和 TLBO 算法的快速收敛能力。

3.2　算法流程

HHSTLBO 算法的具体流程如图 1 所示。

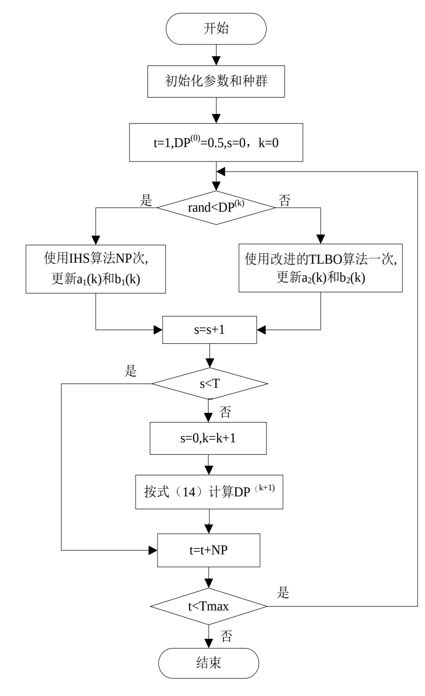


图 1　HHSTL 算法流程

4　实验仿真测试

为检验 HHSTL 算法的性能，将其与另外七种优秀算法（DIHS、HIS、RCGHS、TLBO、ITLBO、MTLBO、TLBO-GC）在 16 个不同类型的标准测试函数上进行仿真测试。其中，F_1～F_3 是单峰函数，F_4 是混合函数，F_5 ~ F_{12} 是复杂的多峰函数，F_{13}～F_{16} 是对 4 个经典的多峰函数进行了 Shift 操作，这些类型各异的标准函数能很好地测试算法的通用性。

F_1：Sphere　Function　　　F_2：Sphere Shift Function
F_3 ：Zakh　Function　　　F_4：Bohachevsky Shift Function
F_5：Ackley　Function　　　F_6：Griewank　Function
F_7：Levy　Function　　　F_8：Schwefel2.22 Function
F_9：Rastrigin　Function　　　F_{10}：Michalewics　Function
F_{11}：Schwefel　Function　　　F_{12}：Schwefel 2.26　Function
F_{13}：Ackley　Shift　Function
F_{14}：Griewank　Shift　Function
F_{15}：Rastrigin　Shift　Function
F_{16}：Schaffer　Shift　Function

4.1　实验环境与参数设置

本文所有测试的环境为：戴尔 PC 机 Intel Core（TM）i7-4790 3.6 GHz CPU，8 GB 内存；Window7 操作系统，MATLAB R2014b 软件。为保证测试的公平，各种算法采用相同的适应度函数评价次数（5000×维数）。比较算法的参数设置参考于原文献，设置具体如表 1 所示。

表 1　参数设置

算法	种群大小	其他
DIHS	10	$HMCR=0.99, PAR_{max}=0.99, PAR_{min}=0.15$ $Bw_{max}=(xU-xL)/20, Bw_{min}=(xU-xL)/(le+15)$ $S_{max}=200/D, S_{min}=20/D, fw_{mid}=(xU-xL)/(le+4)$
IHS	5	$HMCR=0.95, PAR_{max}=0.99, PAR_{min}=0.45$ $Bw_{max}=(xU-xL)/20, Bw_{min}=0.0001$
RCGHS	5	/
TLBO	10	/
ITLBO	40	/
MTLBO	10	$SR_{max}=0.8, SR_{min}=0.3, SFR=0.35$
TLBO-GC	10	/
HHSTL	10	$HMCR=0.99, PAR_{max}=0.99, PAR_{min}=0.1$, $F=0.35,\ Bw_{max}=\lambda_{max}=(xU-xL)/10$ $Bw_{min}=\lambda_{min}=(xU-xL)/(le+15)$ $SCR=0.3, SDR=0.03, T=150$

4.2　实验结果与分析

表 2 统计了 16 个测试函数在 D=100 时的 30 次独立实验结果，包括每个算法的目标函数适应度值的最优值(Best)、平均值(Mean)、最差值（Worst）和方差（Std）。所测试的函数中除 F_{10} 以外，理想最优值均为 0，所以均值越接近 0，表明算法收敛精度越高，而方差越小，表明运算结果的浮动范围越小，算法越稳定。

从表 2 可以看出，RCGHS、TLBO、ITLBO 及 TLBO-GC 算法在四个函数 F_1、F_2、F_6、F_8 能达到理论最优值，但在 F_{11}~F_{16} 上效果明显很差；DIHS 算法在七个函数 F_4、F_5、F_6、F_9、F_{11}、F_{12}、F_{14} 能取得最优值，IHS 算法在三个函数 F_{10}、F_{11}、F_{12} 能取得最优值，但它们在函数 F_1、F_2、F_6、F_8 上收敛精度都不高；而本文所提的 HHSTL 算法在 16 个测试函数上的平均值和方差均能获得最优值，说明所提算法综合了 HS 和 TLBO 两种算法的优点，具有较强的优化能力。为进一步分析实验结果，图 2、3 给出了八种算法 30 次独立运行的平均收敛曲线图和最优值分布盒图（D=100 维）。从收敛曲线图不难看出，本文算法对两个复杂优化函数（Rastrigin Shift Function 、Schaffer　Shift Function)的收敛曲线呈下降趋势，收敛效果明显优于其他算法，说明本文算法的全局搜索能力较强，不易陷入局部最优。从统计盒图可以看出，本文算法的 30 次最优值的分布基本呈一条直线，说明本文算法具有很好的稳定性。

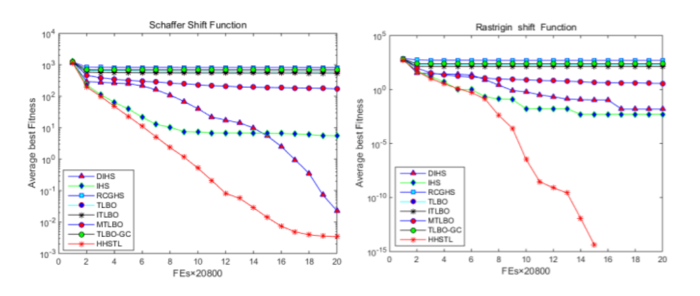


图 2　收敛曲线图

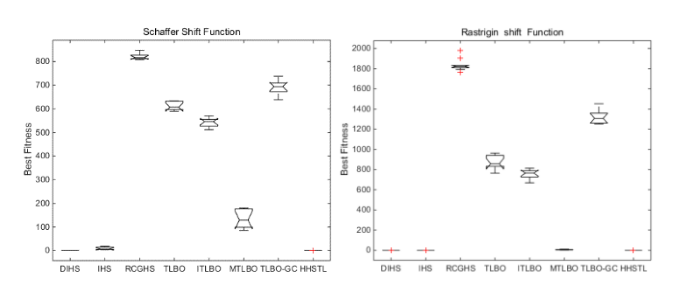


图 3　最优值分布统计盒图

为检验本文算法与比较算法之间的差异性，表 3 列出了本文算法与七种比较算法在 16 个测试问题上 30 次独立运行结果的 Wilcoxon 符号秩检验的 P-value 值，“—”、“+”、“≈”分别表示比较算法与本文算法的成绩相比是差、好和相当（NaN 表示算法成绩无差异）。

从表 3 可以看出，七种比较算法在测试问题上没有优于本文算法的，DIHS 在七个问题上与本文算法相当，IHS 算法在六个问题上与本文算法相当，RCGHS、TLBO 和 ITLBO 算法均在五个问题上与本文算法相当。由此可见，本文算法与比较算法在大多数测试问题上存在显著性差异，具有明显的统计意义。

表 2 算法 30 次独立运行结果比较（D=100）

	ALG	Best	Mean	Worst	Std		ALG	Best	Mean	Worst	Std
F1	DIHS	2.88E-20	3.15E-20	3.61E-20	2.22E-21	F2	DIHS	1.03E-17	1.18E-17	1.30E-17	7.73E-19
	IHS	9.26E-27	1.03E-26	1.12E-26	6.56E-28		IHS	3.44E-24	4.00E-24	4.77E-24	4.47E-25
	RCGHS	**0**	**0**	**0**	**0**		RCGHS	**0**	**0**	**0**	**0**
	TLBO	**0**	**0**	**0**	**0**		TLBO	**0**	**0**	**0**	**0**
	ITLBO	**0**	**0**	**0**	**0**		ITLBO	**0**	**0**	**0**	**0**
	MTLBO	2.24E-84	2.62E-82	1.86E-81	5.75E-82		MTLBO	1.07E-81	1.48E-80	4.72E-80	1.46E-80
	TLBO-GC	**0**	**0**	**0**	**0**		TLBO-GC	**0**	**0**	**0**	**0**
	HHSTL	**0**	**0**	**0**	**0**		HHSTL	**0**	**0**	**0**	**0**
F3	DIHS	2.41E+02	2.80E+02	3.18E+02	3.12E+01	F4	DIHS	**0**	**0**	**0**	**0**
	IHS	1.40E+02	1.65E+02	1.98E+02	1.94E+01		IHS	**0**	3.26E-04	3.26E-03	1.03E-03
	RCGHS	3.57E+13	3.87E+14	7.43E+14	2.33E+14		RCGHS	4.87E+03	4.92E+03	4.99E+03	3.50E+01
	TLBO	1.40E-254	9.32E-235	9.31E-234	**0**		TLBO	5.27E+01	1.62E+02	2.64E+02	7.58E+01
	ITLBO	1.73E-110	4.17E-102	4.15E-101	1.31E-101		ITLBO	2.41E+01	3.30E+01	4.29E+01	5.94E+00
	MTLBO	1.31E+02	1.91E+02	2.57E+02	3.41E+01		MTLBO	**0**	1.80E-14	1.28E-13	3.98E-14
	TLBO-GC	1.63E-179	8.51E-153	8.50E-152	2.69E-152		TLBO-GC	2.98E+03	3.40E+03	3.69E+03	2.28E+02
	HHSTL	**0**	**1.49E-262**	**1.49E-261**	**0**		HHSTL	**0**	**0**	**0**	**0**
F5	DIHS	4.33E-10	4.48E-10	4.67E-10	1.10E-11	F6	DIHS	**0**	**0**	**0**	**0**
	IHS	2.28E-13	2.51E-13	2.71E-13	1.15E-14		IHS	0	7.40E-04	7.40E-03	2.34E-03
	RCGHS	4.44E-15	4.80E-15	7.99E-15	**1.12E-15**		RCGHS	**0**	**0**	**0**	**0**
	TLBO	4.44E-15	6.93E-15	7.99E-15	1.72E-15		TLBO	**0**	**0**	**0**	**0**
	ITLBO	4.44E-15	6.93E-15	7.99E-15	1.72E-15		ITLBO	**0**	**0**	**0**	**0**
	MTLBO	5.06E-14	5.84E-14	6.48E-14	4.37E-15		MTLBO	0	4.395E-13	4.394E-12	1.39E-12
	TLBO-GC	4.44E-15	6.57E-15	7.99E-15	1.83E-15		TLBO-GC	**0**	**0**	**0**	**0**
	HHSTL	**8.88E-16**	**1.24E-15**	**4.44E-15**	**1.12E-15**		HHSTL	**0**	**0**	**0**	**0**
F7	DIHS	5.29E-20	6.02E-20	6.44E-20	3.77E-21	F8	DIHS	1.085E-09	1.135E-09	1.178E-09	2.927E-11
	IHS	1.68E-26	2.01E-26	2.38E-26	2.45E-27		IHS	7.359E-13	7.83E-13	8.492E-13	3.617E-14
	RCGHS	9.39E+00	9.49E+00	9.57E+00	5.01E-02		RCGHS	**0**	**0**	**0**	**0**
	TLBO	4.15E+00	5.18E+00	6.18E+00	5.52E-01		TLBO	**0**	**0**	**0**	**0**
	ITLBO	4.12E+00	4.62E+00	5.55E+00	4.47E-01		ITLBO	**0**	**0**	**0**	**0**
	MTLBO	1.24E-23	5.67E-22	3.11E-21	9.24E-22		MTLBO	4.207E-46	3.448E-45	1.189E-44	4.129E-45
	TLBO-GC	7.92E+00	8.21E+00	8.55E+00	2.26E-01		TLBO-GC	**0**	**0**	**0**	**0**
	HHSTL	**4.86E-27**	**7.15E-27**	**9.04E-27**	**1.42E-27**		HHSTL	**0**	**0**	**0**	**0**
F9	DIHS	**0**	**0**	**0**	**0**	F10	DIHS	-9.93E+01	-9.93E+01	-9.91E+01	5.48E-02
	IHS	**0**	3.04E-03	1.30E-02	5.05E-03		IHS	**-9.94E+01**	**-9.94E+01**	**-9.93E+01**	4.35E-02
	RCGHS	**0**	**0**	**0**	**0**		RCGHS	-4.09E+01	-3.80E+01	-3.06E+01	3.05E+00
	TLBO	**0**	8.56E+00	8.56E+01	2.71E+01		TLBO	-9.03E+01	-7.90E+01	-5.07E+01	1.12E+01
	ITLBO	**0**	**0**	**0**	**0**		ITLBO	-8.48E+01	-7.83E+01	-6.64E+01	6.04E+00
	MTLBO	4.80E+00	8.77E+00	1.20E+01	1.97E+00		MTLBO	-9.65E+01	-9.61E+01	-9.57E+01	2.71E-01
	TLBO-GC	**0**	1.12E+02	5.00E+02	1.88E+02		TLBO-GC	-4.79E+01	-4.27E+01	-3.70E+01	3.33E+00
	HHSTL	**0**	**0**	**0**	**0**		HHSTL	**-9.94E+01**	**-9.94E+01**	**-9.93E+01**	**4.24E-02**
F11	DIHS	**1.31E-10**	**1.31E-10**	**1.31E-10**	**0**	F12	DIHS	**1.31E-10**	**1.31E-10**	**1.31E-10**	**0**
	IHS	**1.31E-10**	**1.31E-10**	**1.31E-10**	**0**		IHS	**1.31E-10**	**1.31E-10**	**1.31E-10**	**0**
	RCGHS	3.74E+04	3.84E+04	3.94E+04	7.30E+02		RCGHS	3.72E+04	3.82E+04	3.90E+04	6.13E+02
	TLBO	1.52E+04	1.95E+04	2.28E+04	2.52E+03		TLBO	1.09E+04	1.89E+04	2.55E+04	4.90E+03
	ITLBO	1.77E+04	1.94E+04	2.19E+04	1.19E+03		ITLBO	1.41E+04	1.74E+04	2.04E+04	1.66E+03
	MTLBO	1.60E-10	2.68E-10	4.58E-10	9.23E-11		MTLBO	1.60E-10	2.44E-10	4.07E-10	7.87E-11
	TLBO-GC	2.65E+04	2.84E+04	3.20E+04	1.58E+03		TLBO-GC	2.55E+04	2.74E+04	2.91E+04	1.19E+03
	HHSTL	**1.31E-10**	**1.31E-10**	**1.31E-10**	**0**		HHSTL	**1.31E-10**	**1.31E-10**	**1.31E-10**	**0**
F13	DIHS	4.19E-10	4.44E-10	4.61E-10	1.38E-11	F14	DIHS	**0**	**0**	**0**	**0**
	IHS	2.42E-13	2.54E-13	2.70E-13	**1.01E-14**		IHS	**0**	1.11E-17	1.11E-16	3.51E-17
	RCGHS	2.06E+01	2.08E+01	2.10E+01	9.15E-02		RCGHS	2.75E+03	3.19E+03	4.68E+03	7.59E+02
	TLBO	1.89E+01	1.91E+01	1.93E+01	1.33E-01		TLBO	3.26E+00	6.89E+01	1.45E+02	4.95E+01
	ITLBO	1.77E+01	1.84E+01	1.90E+01	4.06E-01		ITLBO	1.03E-02	8.68E-01	4.51E+00	1.44E+00
	MTLBO	3.13E-11	1.13E-10	2.36E-10	6.76E-11		MTLBO	4.44E-16	7.58E-04	7.41E-03	2.34E-03
	TLBO-GC	1.99E+01	2.01E+01	2.04E+01	1.82E-01		TLBO-GC	1.62E+03	1.90E+03	2.12E+03	1.60E+02
	HHSTL	**1.99E-13**	**2.18E-13**	**2.45E-13**	**1.01E-14**		HHSTL	**0**	**0**	**0**	**0**
F15	DIHS	**0**	3.33E-03	3.19E-02	1.01E-02	F16	DIHS	2.38E-03	6.59E-03	1.77E-02	6.73E-03
	IHS	**0**	8.42E-03	5.75E-02	1.81E-02		IHS	4.05E+00	9.30E+00	1.84E+01	5.41E+00
	RCGHS	1.77E+03	1.84E+03	1.98E+03	6.08E+01		RCGHS	8.09E+02	8.21E+02	8.48E+02	1.24E+01
	TLBO	7.64E+02	8.73E+02	9.64E+02	6.52E+01		TLBO	5.90E+02	6.12E+02	6.34E+02	1.72E+01
	ITLBO	6.69E+02	7.52E+02	8.14E+02	5.03E+01		ITLBO	5.12E+02	5.42E+02	5.71E+02	2.01E+01
	MTLBO	2.00E+00	5.59E+00	1.10E+01	2.75E+00		MTLBO	8.60E+01	1.35E+02	1.80E+02	3.96E+01
	TLBO-GC	1.25E+03	1.32E+03	1.45E+03	6.40E+01		TLBO-GC	6.38E+02	6.92E+02	7.38E+02	2.75E+01
	HHSTL	**0**	**5.86E-15**	**3.91E-14**	**1.32E-14**		HHSTL	**4.59E-05**	**3.32E-03**	**1.67E-02**	**5.33E-03**

表 3 Wilcoxon 符号秩检验的 P-value（D=100）

函数	DIHS	IHS	RCGHS	TLBO	ITLBO	MTLBO	TLBO-GC
F1	6.39E-05	6.39E-05	NaN	NaN	NaN	6.39E-05	NaN
F2	6.39E-05	6.39E-05	NaN	NaN	NaN	6.39E-05	NaN
F3	1.79E-04	1.79E-04	1.79E-04	1.79E-04	1.79E-04	1.79E-04	1.79E-04
F4	6.16E-05	1.71E-03	1.78E-04	1.78E-04	1.78E-04	2.72E-01	1.78E-04
F5	8.74E-05	8.45E-05	1.24E-04	9.14E-05	9.14E-05	7.57E-05	1.17E-04
F6	NaN	3.68E-01	NaN	NaN	NaN	7.79E-02	NaN
F7	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
F8	6.39E-05	6.39E-05	NaN	NaN	NaN	6.39E-05	NaN
F9	NaN	1.49E-02	NaN	3.68E-01	NaN	6.39E-05	7.79E-02
F10	1.71E-03	2.12E-01	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
F11	NaN	NaN	6.39E-05	6.39E-05	6.39E-05	6.29E-05	6.39E-05
F12	NaN	NaN	6.39E-05	6.39E-05	6.39E-05	6.34E-05	6.39E-05
F13	1.71E-03	2.12E-01	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
F14	NaN	3.68E-01	6.39E-05	6.39E-05	6.39E-05	6.39E-05	6.39E-05
F15	4.56E-01	8.37E-03	1.11E-04	1.11E-04	1.11E-04	1.11E-04	1.11E-04
F16	1.21E-01	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
—	9	10	11	11	11	14	13
+	0	0	0	0	0	0	0
≈	7	6	5	5	5	2	3

4.3 选择周期 T 的分析

混合算法经过 T 次迭代后需要根据动态选择概率来重新确定下一周期中两种算法作为种群更新方式的比例。为检验选择周期 T 对算法的影响，在六个测试函数上对 T 的不同取值进行了测试（D=30）。表 4 统计了 30 次独立实验的最优目标函数适应度值的平均值，图 4 给出了不同 T 值下 30 次测试的分布盒图。

从表 4 可以看出，除函数 Griewank shift Function 外，多数测试函数在不同 T 值下得到的平均适应度值差异不是很大，说明所提算法对此参数不太敏感。根据表 4 和图 4 的结果，综合考虑解的精度及分布统计情况，本文取 T=150。

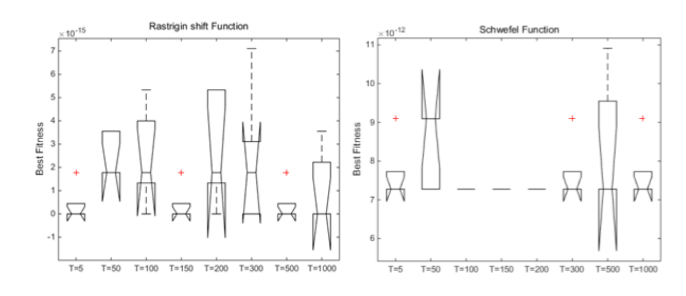


图 4 不同 T 值下的最优值分布统计盒图

表 4 不同 T 值下 30 次独立实验的最优目标函数的平均适应值

Function	T=5	T=50	T=100	T=150	T=200	T=300	T=500	T=1000
Schwefel Function	8.37E-12	**7.28E-12**	**7.28E-12**	**7.28E-12**	7.64E-12	8.37E-12	7.64E-12	7.64E-12
Levy Function	1.07E-29	9.05E-30	1.21E-29	**1.57E-30**	1.63E-29	3.06E-29	1.75E-29	1.32E-29
Schwefel2.26 Function	8.37E-12	**7.28E-12**	**7.28E-12**	**7.28E-12**	7.64E-12	8.37E-12	7.64E-12	7.64E-12
Ackley shift Function	7.18E-14	6.59E-14	7.59E-14	**6.25E-14**	6.79E-14	6.32E-14	6.84E-14	6.47E-14
Rastrigin shift Function	**3.55E-16**	2.49E-15	2.49E-15	**3.55E-16**	2.84E-15	2.13E-15	3.55E-16	1.07E-15
Griewank shift function	**0**	4.93E-03	**0**	**0**	6.40E-03	8.37E-03	3.94E-03	7.87E-03

4.4 动态选择策略分析

为进一步分析本文算法所设计的动态选择策略，分别选择了单峰和复杂多峰函数进行了测试。图 5 给出了两种不同类型函数的动态选择概率变化曲线图。

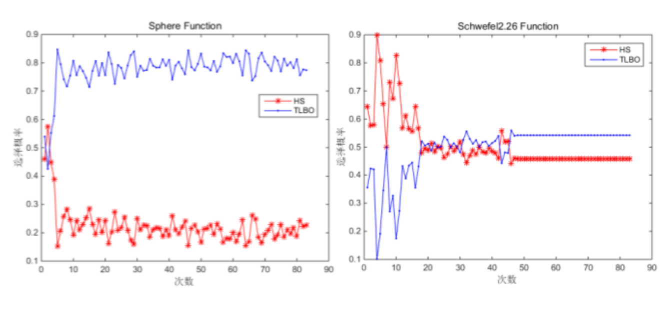


图 5 动态选择概率变化曲线图

从图 5 中可以看出，动态选择策略能够根据优化问题的特点和迭代过程中两种优化算法的具体表现动态地选择合适的更新策略来满足不同进化阶段的需求。对 Sphere Function 这样的单峰函数，TLBO 算法因为良好的收敛能力，因而在整个优化过程中占据主导地位，选择概率明显高于 HS 算法；对 Schwefel2.26 Function 这样的复杂多峰函数，由于存在多个局部极小值点，进化的前期需要进行全局搜索，HS 算法因为较强的全局搜索能力而获得较高的选择比例，能快速定位最优解的有效区域。随着迭代的进行，两种算法的选择概率慢慢趋近，交替变化，在迭代的后期，TLBO 算法的选择概率增大，主要来进行精细搜索，提高解的精度。迭代过程中，两种算法都保持着一定的选择比例，HS 算法能够维持种群的多样性，而 TLBO 算法能够加快收敛速度，提高解的精度。

5 结束语

为提高对多种不同类型问题的优化能力，本文提出了一种和声教与学混合优化算法——HHSTL。算法使用由种群活跃率及种群最优个体更新率构成的动态选择概率将两种算法进行了有效地融合。通过在 16 个不同类型的 Benchmark 函数上对 HHSTL 算法及其他七种优秀算法进行仿真比较，统计结果显示，所提算法汲取了 HS 和 TLBO 两种算法的长处，优势互补，求解精度高，稳定性好，能在较多不同类型的优化问题上取得良

好的效果。但因为 HHSTL 算法是由 IHS 和 TLBO 两种算法混合而成，参数较多，需要进行大量的测试。在后续研究中，可以设计有效的参数自适应策略来进行改进，也可以将该算法应用到实际的工程优化问题中，检验其具体实用性。

参考文献：

[1] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: harmony search [J]. Simulation, 2001, 2 (2): 60-68.

[2] Lee K S, Geem Z W. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice [J]. Computer Methods in Applied Mechanics & Engineering, 2005, 194 (36): 3902-3933.

[3] Wu Bin, Qian Cunhua, Ni Weihong, *et al.* Hybrid harmony search and artificial bee colony algorithm for global optimization problems [J]. Computers & Mathematics with Applications, 2012, 64 (8): 2621-2634.

[4] Rao R V, Savsani V J, Vakharia D P. Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems [J]. Information Sciences, 2012, 183 (1): 1-15.

[5] Rao R V, Patel V. Multi-objective optimization of two stage thermoelectric cooler using a modified teaching-learning based optimization algorithm [J]. Engineering Applications of Artificial Intelligence, 2013, 26 (1): 430-445.

[6] Pickard J K, Carretero J A, Bhavsar V C. On the convergence and origin bias of the teaching-learning-based optimization algorithm [J]. Applied Soft Computing, 2016, 46 (9): 115-127.

[7] Aydogdu I, Akin A. Optimum design of steel space frames by hybrid teaching-learning based optimization and harmony search algorithms [C]// Proc of the 17th International Conference on Structural Engineering. 2015. 2015: 1486-1493.

[8] Tuo Shouheng, Yong Longquan, Deng Fangan, *et al.* HSTLBO: a hybrid algorithm based on harmony search and teaching-learning-based optimization for complex high-dimensional optimization problems [J]. Plos One, 2017, 12 (4): 1-23.

[9] Yong Longquan, Tuo Shouheng, Gao Kai. Harmony search with teaching-learning algorithm for nonnegative linear least square [J]. Journal of Interdisciplinary Mathematics, 2017, 20 (8): 1661-1677.

[10] Ouyang Haibin, Ma Ge, Liu Guiyun, *et al.* Hybrid teaching-learning based optimization with harmony search for engineering optimization problems [C]// Proc of the 36th Chinese Control Conference. Dalian: IEEE Press, 2017: 2714-2717.

[11] Herrera F, Lozano M, Molina D. Test suite for of soft computing on scalability of evolutionary other metaheuristics for large scale continuous problems [EB/OL]. (2010) . http://sci2s. ugr. es/eamhco/CFP. php.

[12] Zhao Shizheng, Suganthan P N, Das S. Self-adaptive differential evolution with modified multi-trajectory search for CEC'2010 large scale optimization [J]. Lecture Notes in Computer Science, 2010, 15 (11): 1-10.

[13] Tuo Shouheng, Zhang Junying, Yong Longquan, *et al.* A harmony search algorithm for high-dimensional multimodal optimization problems [J]. Digital Signal Processing, 2015, 46: 151-163.

[14] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems [J]. Applied mathematics and computation, 2007, 188 (2): 1567-1579.

[15] 翟军昌，秦玉平. 随机交叉全局和声搜索算法 [J]. 计算机工程与应用, 2018, 54 (12): 21-26. (Zhai Junchang, Qin Yuping. Random crosser global harmony search algorithm [J]. Computer Engineering and Applications, 2018, 54 (12): 21-26.)

[16] Chen Debao, Zou Feng, Wang Jiangtao, *et al.* An improved teaching-learning based optimization algorithm for solving global optimization problem [J]. Information Sciences, 2015, 297: 171-190.

[17] 拓守恒，雍龙泉，黎延海，等. 面向高维复杂多模态问题的教与学优化求解算法 [J]. 计算机应用研究, 2017, 34 (7): 1939-1945. (Tuo Shouheng, Yong Longquan, Li Yanhai, *et al.* Teaching-learning-based optimization algorithm for solving high-dimensional complex optimization problems with multimodality [J]. Application Research of Computers, 2017, 34 (7): 1939-1945.)

[18] Ouyang Haibin, Gao Liqun, Kong Xiangyong, *et al.* Teaching-learning based optimization with global crossover for global optimization problems [J]. Applied Mathematics and Computation, 2015, 265 (C): 533-556.